Design of a Secure Mutually Authenticated Key-Agreement Protocol for Multi-server Architecture

Alavalapati Goutham Reddy\textsuperscript{1}, Eun-Jun Yoon\textsuperscript{2}, Young-Ju Kim\textsuperscript{1}, Kee-Young Yoo\textsuperscript{1}\textsuperscript{*}

\textsuperscript{1} School of Computer Science and Engineering, Kyungpook National University, Daegu, Republic of Korea.
\textsuperscript{2} Department of Cyber Security, Kyungil University, Gyeongbuk, Republic of Korea.

* Corresponding author. Email: yook@knu.ac.kr
Manuscript submitted January 5, 2017; accepted April 20, 2017.
doi: 10.17706/jcp.13.2.194-203

Abstract: Authentication with key-agreement protocols for multi-server architecture are emerging as a solution to conquer the traditional client-server architecture's limitations such as repeated registrations with distinct tokens and credentials. Since Li et al.’s first proposed authentication protocol for multi-server architecture, several similar protocols have followed this queue. Majority of these protocols have been designed while the users sharing their plain or digested credentials with the servers during either registration or authentication phases. This weakens the security by making it vulnerable to severe security threats called privileged insider attacks, user impersonation attacks and server impersonation attacks. To overcome the aforementioned problems, this paper forwards an authentication with key-agreement protocol for multi-server architecture based on biometrics. The proposed protocol is absolutely light-weight due to its design mainly based on one-way hash function. The analysis section of this paper shows that the proposed protocol performs better than related protocols and makes it suitable for practical applications.

Key words: Authentication, key-agreement, protocol, multi-server, three-factor, security, performance.

1. Introduction

The vast expansion of internet and ubiquitous computing technologies have necessitated the authentication of every remote user. Cryptographic authentication is a secure practice of transferring credentials to determine someone, in fact, who they are proclaimed to be and providing authorization to access the services subsequently. Typical authentication can be obtained in distinctive ways namely knowledge factors (passwords), possession factors (tokens) and inherence factors (biometrics) are some well-known methods. Several authors designed authentication protocols for multi-server environment using either two of the above factors or all the three factors [1]-[22]. This paper discusses the recently proposed three-factor authentication protocols under the hypothesis of biometrics are more robust than passwords and smartcards.

phases. In 2014, Chuang et al. [5] proposed an anonymous three-factor multi-server authenticated key agreement protocol. Their protocol is constructed mainly on one-way hash function which makes it suitable for real-time applications. Unfortunately, Mishra et al. [6] in 2014 & Lin et al. [7] in 2015 pointed out several weaknesses of Chuang et al.'s protocol such as lack of user anonymity, susceptible to server spoofing attacks, stolen smartcard attacks, user impersonation attacks, denial-of-service attacks, and session-key compromise. Then they proposed an improved protocol over Chuang et al.'s protocol. However, Lu et al. [8] in 2015 & Wang et al. [9] in 2016 stated that Mishra et al.'s protocol is exposed to user and server masquerading attacks, replay attacks, forgery attacks, denial-of-service attacks, lack of perfect forward secrecy and user anonymity. Lu et al. [8] in 2015 & Wang et al. [9] in 2016 proposed improved protocols over Mishra et al.'s protocol. In 2015, Jiang et al. [10], He et al. [11] & Odulu et al. [12] put forward various authenticated key-agreement protocols for multi-server environment based on biometrics. Above three protocols involve registration center in the authentication phase which makes them inefficient due to overwhelming calculations at registration center. Additionally, Odulu et al. asserted that He et al.'s protocol has drawbacks in login and password change phases and is also prone to impersonation attacks. Jiang et al.'s protocol cannot provide user anonymity and makes it prone to trace attacks. In 2016, Reddy et al. [13] cryptanalyzed Lu et al.'s protocol and showed the weaknesses such as prone to impersonation attacks, man-in-middle attacks, clock synchronization problem, lack of user anonymity and lack of perfect forward secrecy. Then they proposed an improved robust protocol built on elliptic curve cryptography. Most recently, Wang et al. proposed another three-factor authenticated key agreement protocol for multi-server environment. However, their protocol also shown the flaws such as lack of anonymity, vulnerability to impersonation attacks, insider attacks and clock synchronization problems.

<table>
<thead>
<tr>
<th>Table 1. Notations of the Proposed Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_i$</td>
</tr>
<tr>
<td>$AS$</td>
</tr>
<tr>
<td>$RS$</td>
</tr>
<tr>
<td>$ID_{U_i}$</td>
</tr>
<tr>
<td>$PW_{U_i}$</td>
</tr>
<tr>
<td>$BIO_{U_i}$</td>
</tr>
<tr>
<td>$r_{U_i}$</td>
</tr>
<tr>
<td>$SID_{AS}$</td>
</tr>
<tr>
<td>$USK$</td>
</tr>
<tr>
<td>$ASK$</td>
</tr>
<tr>
<td>$N_1, N_2$</td>
</tr>
<tr>
<td>$T_{reg}$</td>
</tr>
<tr>
<td>$C_u$</td>
</tr>
<tr>
<td>$SK$</td>
</tr>
<tr>
<td>$A$</td>
</tr>
<tr>
<td>$SC$</td>
</tr>
<tr>
<td>$Gen(BIO, \phi)$</td>
</tr>
<tr>
<td>$Rep(BIO, \phi)$</td>
</tr>
<tr>
<td>$h(\cdot)$</td>
</tr>
<tr>
<td>$\oplus$</td>
</tr>
<tr>
<td>$|$</td>
</tr>
</tbody>
</table>

Our contributions: The keen observation of above related works clearly proves that most of them are vulnerable to impersonation attacks. Therefore, this paper proposes another three-factor authentication protocol for multi-server architecture. Unlike the existing protocols, the proposed protocol attains perfect mutual authentication without sharing its user's credentials throughout any phase. This nature of proposed protocol resists all sorts of attacks and makes it robust compared to other protocols.

Roadmap of the paper—Section 2 presents the proposed scheme. Section 3 portrays security analysis of
the proposed scheme in detail. Section 4 affords performance analysis. At last, Section 5 concludes the paper.

2. The Proposed Protocol

This section proposes three-factor remote mutual authentication with key agreement protocol for multi-server architecture. The proposed protocol comprises three participants: user (Ui), application server (AS), registration server (RS) and seven phases: application server registration phase, user registration phase, login phase, mutual authentication with key agreement phase, password and biometrics changing phase, dynamic server addition phase, and user revocation/re-registration phase. The various notations used in the proposed protocol are listed in Table 1.

2.1. Application Server Registration Phase

In this phase, a new AS sends a registration request to the RS in order to become an authorized server of the network. The AS registration process occurs via a secure channel with following steps:

Step 1: AS sends registration request < SIDA > to the RS.

Step 2: RS computes KS = h(SIDA || ASK) and RS stores {SIDA KS} in its database table T5.

Step 3: RS sends < KS h(ASK) > to AS, which can be used in further phases of authentication.

2.2. User Registration Phase

A new user Ui who desires to avail the services provided by any AS must register with RS. Assume that Ui obtains a SC with the value {h(·)} upon formal request to RS. Ui goes after the following steps to register with RS via a secure channel as shown in Fig. 1.

Step 1: Ui chooses IDU, PWU, and generates a random number ru ∈ Z∗. Ui computes PIDU = h(IDU || ru), PWDU = h(PWU || ru) and sends a request message < PIDU, PWDU > to RS.

Step 2: RS verifies the duplication of PIDU and then computes Q5 = h(PIDU || K5), R5 = Q5 ⊕ PWDU for all the registered SIDA and stores {SIDA, R5} in the table T5. RS also stores {PIDU, C5, T5 = 1} in the table T5, where T5 = 1 means Ui is registered once and is in active state. RS computes W5 = h(PIDU || USK) and personalizes {W5, T5, h(ASK)} into the SC to be delivered to Ui.

Step 3: Ui scans his/her BIOU at the provided sensor with card reading machine and computes X5 = W5 ⊕ PWDU, C5 = h(IDU || W5), (σU, θU) = Gen(BIOU), and V5 = ru ⊕ h(σU). Ui replaces W5 with X5 and stores {C5, V5, θU} on SC. Thus the SC finally contains the parameters SC{X5, V5, C5, T5, θU, h(·), h(ASK)}.
2.3. Login Phase

When a \( U_i \) wants to access the services of any registered \( AS \), he/she can launch the login request by inserting \( ID_U, PW_U \) and \( BIO_U \) as detailed in the Fig. 2.

Step 1: \( SC \) computes \( au' = \text{Rep}(BIO_U, \theta_U) \), \( ru = V_U \oplus h(au') \), \( PID_U = h(ID_U || ru) \), \( PWD_U = h(PW_U || ru) \), \( W_5 = X_5 \oplus PWD_U \) and then verifies whether the condition \( C_U \equiv h(ID_U || W_5) \) holds. If it generates a negative result, the login request can be terminated. Otherwise, the list of \( AS \) appears on the card reading machine.

Step 2: \( SC \rightarrow AS: M_2 = < B_{US}, D_{US}, N_2 > \)

\( U_i \) selects the \( AS \) he wanted to communicate, then \( SC \) retrieves corresponding \( AS \)'s \( R_5 \) value from \( T_U \) and extracts \( Q_5 = R_5 \oplus PWD_U \). \( SC \) generates \( N_1 \in Z_p^* \) and computes \( Q_5 = R_5 \oplus PWD_U \) \( B_{US} = PID_U \oplus h(SID_A || N_1 || h(ASK)) \). \( D_{US} = h(PID_U || Q_5 || N_2) \). \( SC \) launches the login request message \( M_1 = < B_{US}, D_{US}, N_1 > \) to \( AS \).

2.4. Mutual Authentication with Key-Agreement Phase

During this phase, \( U_i \) and \( AS \) authenticates each other and computes a session key for further secure communication over public channel. The entire mutual authentication with key agreement phase is illustrated in Fig. 2.

![Fig. 2. Summary of login and mutual authentication phases.](image)

Step 1: \( AS \) computes \( PID_U = B_{US} \oplus h(SID_A || N_1 \| h(ASK)) \), \( Q_5 = h(PID_U || K_3) \) and verifies the condition \( D_{US} \equiv h(PID_U || Q_5 || N_2) \). If the condition holds, then \( AS \) can authenticate \( U_i \) otherwise the process can be terminated.

Step 2: \( AS \rightarrow SC: M_2 = < E_{US}, N_2 > \)

\( AS \) generates \( N_2 \in Z_p^* \) and computes \( SK = h(Q_5 \| N_1 \| N_2 \| PID_U) \), \( E_{US} = h(SK \| SID_A \| N_1 \| N_2) \), then and sends \( M_2 = < E_{US}, N_2 > \) to \( SC \).

Step 3: \( SC \rightarrow AS: M_3 = < F_{US} > \)

\( SC \) computes \( SK = h(Q_5 \| N_1 \| N_2 \| PID_U) \) and verifies the condition \( E_{US} \equiv h(SK \| SID_A \| N_1 \| N_2) \). If the condition holds, then \( U_i \) can authenticate \( AS \), otherwise the process can be terminated. \( SC \) computes \( F_{US} = h(SID_A \| N_1 \| N_2 \| SK) \) and sends it to \( AS \).

Step 4: \( AS \) verifies \( F_{US} \equiv h(SID_A \| N_1 \| N_2 \| SK) \) and reconfirms the authenticity of \( U_i \). Now, \( U_i \) and \( AS \) are set to start the communication with the computed session key \( SK \).
2.5. Password and Biometrics Changing Phase

This procedure is invoked when $U_i$ wish to update his/her existing password or biometrics with new ones. In this procedure, $U_i$ can change his password or biometrics by inserting SC and inputting $ID_{U_i}\), PW_{U_i}$ and $BIO_{U_i}$ over a secure channel without the help of RS as follows.

Step 1: $SC$ computes $o' = Rep(BIO_{U_i}, \theta_0)$, $r_0 = V_0 \oplus h(o')$, $PID_U = h(ID_{U_i} | r_0)$, $PWD_U = h(PW_{U_i} | r_0)$, $W_s = X_s \oplus PWD_U$ and then verifies whether the condition $C_U \neq h(ID_{U_i} | W_s)$ holds. If it generates a negative result, the login request can be terminated.

Step 2: $SC$ retrieves $R_S = Q_S \oplus PWD_U$ for all $SID_A$ using the current $PW_{U_i}$. $U_i$ chooses a new password $PW_{U_i}^{new}$ and then computes $PWD_{U_i}^{new} = h(PW_{U_i}^{new} | r_0)$, $X_{s}^{new} = W_s \oplus PWD_{U_i}^{new}$, $(o^{new}, \theta_0^{new}) = Gen(BIO_{U_i}^{new}), V_{i0}^{new} = v_0 \oplus h(o^{new}), R_{s}^{new} = Q_s \oplus PWD_{U_i}^{new}$ for all $SID_A$.

Step 3: $U_i$ updates the table $T_i^{new}$ and the parameters $X_{s}^{new}, V_{i0}^{new}, \theta_0^{new}$ on the $SC$. Thus, the $SC$ finally consists of the parameters $SC\{X_{s}^{new}, V_{i0}^{new}, C, T_i^{new}, \theta_0^{new}, h(\cdot), h(ASK)\}$. 

2.6. Dynamic Addition of Application Server Phase

In this phase, a new application server $AS^{new}$ can join the existing network by sending a registration request to the $RS$ in order to become an authorized server. The new application server’s information will be forwarded to the existing users of the network periodically using their stored $(PID_U, C_i)$ or upon the request for updates from $U_i$. In any of the above two cases, $U_i$ is expected to pass through login phase successfully. The $AS^{new}$ registration process consists of following steps.

Step 1: $AS^{new}$ sends registration request <$SID_A^{new}$> to the $RS$.

Step 2: $RS$ computes $K_{new}^{new} = h(SID_A^{new} | ASK)$, and $RS$ stores $(SID_A^{new}, K_{new}^{new})$ in $T_S$.

Step 3: $RS$ sends <$K_{new}^{new}$> to $AS^{new}$ via a secure channel, which can be used in further phases of authentication.

Step 4: $RS$ retrieves the $(PID_{U_i}, C_i)$ of users and computes $Q_{S}^{new} = h(PID_{U_i} | K_{new}^{new})$, and then delivers it to $U_i$ via a secure channel. $RS$ updates $(PID_{U_i}, C_i)$ in the table $T_C$. Upon receiving the new $AS$’s information, $U_i$ computes $R_{S}^{new} = Q_{S}^{new} \oplus PWD_U$ and adds it to their $T_U$.

2.7. User Revocation/Re-Registration Phase

This phase directs $U_i$ when he/she wants revoke the available services or re-register with different identity. In any of the above two cases, $U_i$ is expected to pass through login phase successfully. Both the phases occurs via a secure channel as explained below.

Revocation: During the revocation phase, $U_i$ proves his/her legitimacy and submits the acquired $SC$ to the $RS$. Upon receiving the request, $RS$ sets $T_R = 0$ and updates $(PID_{U_i}, C_i, T_R = 0)$ in the table $T_C$, where $T_R = 0$ means $U_i$ is revoked and inactive. When $U_i$ wishes to resume his/her previous services, then $U_i$ must prove his/her $PID_{U_i} = h(ID_{U_i} | r_0), PWD_{U_i} = h(PW_{U_i} | r_0)$ by retrieving $Q_{S} = R_S \oplus PWD_U$ using the last recent $ID_{U_i}, PW_{U_i}$ and $BIO$. If $U_i$ holds the correct credentials, then $RS$ resumes $U_i$’s services and updates $(PID_{U_i}, C_i, T_R = 1)$ in the table $T_C$.

Re-registration: When $U_i$ wants to re-register with the new credentials, then he/she must prove his/her legitimacy and send the request to $RS$. Upon receiving the request, $RS$ follows the steps described in user registration phase and updates $(PID_{U_i}, C_i, T_R = T_R + 1)$ in the table $T_C$.

3. Security Analysis

This section demonstrates the security analysis of the proposed protocol by describing each security feature. The main aim of the proposed protocol is to increase the degree of security of communicating messages over public or insecure channels.
Proposition 1. The proposed protocol achieves user anonymity and untraceability.

Proof. The transmitted messages \(< M_1, M_2, M_3 >\) between \(U\) and \(AS\) during the login and authentication phases are arbitrary for each session due to their association with the random numbers \(N_1\) and \(N_2\). \(U\)'s original identity \(ID_U\) is encapsulated in the form of \(B_{US} = PID_U \oplus h(SID_1 || N_1 || h(ASK))\), where \(PID_U = h(ID_U || r_0)\). The similar approach is followed in case of all the parameters \(< B_{US}, D_{US}, N_1 >, < E_{US}, N_2 >, \) and \(< F_{US} >\) and accomplished user anonymity. The proposed protocol also provides another important feature called untraceability. The randomness of all the parameters makes it unidentifiable and untraceable to the adversaries.

Proposition 2. The proposed protocol is secure against replay attacks.

Proof a. \(\mathcal{AE}\) may try to establish a new session while impersonating a valid user by replaying the previous transmitted message \(< M_1 >\). However, the proposed protocol can withstand replay attacks using random number \(N_1\) as explained here. During the login and mutual authentication phase, \(AS\) receives the message \(< B_{US}, D_{US}, N_1 >\) and stores the pair \(\{PID_U, N_1\}\) in its database. If \(\mathcal{AE}\) replays the same message \(< M_1^{\mathcal{AE}} >\), \(AS\) retrieves \(\{PID_U, N_1^{\mathcal{AE}}\}\) and compares with the stored \(\{PID_U, N_1\}\). When \(AS\) finds \(N_1^{\mathcal{AE}} = N_1\), then it drops the request and terminate the process. \(U\) and \(AS\) follows the similar way to defy the replay attack on response messages \(< E_{US}, N_2 >\) and \(< F_{US} >\).

Proposition 3. The proposed protocol is secure against stolen smartcard attacks.

Proof. With the hypothesis that \(\mathcal{AE}\) can read a SC stored values using various methods as discussed in, this section describes the resistance of the proposed protocol to stolen smartcard attack. Assume that \(\mathcal{AE}\) is able to read the stored parameters \(\{X_0, V_0, C_0, T_0, \theta_0, h(.), h(ASK)\}\) on a stolen legitimate SC. Now, \(\mathcal{AE}\) may try either launching an authentication request to gain the access to \(AS\) or try deriving actual \(U\)'s credentials from the extracted parameters. However, \(\mathcal{AE}\) undeniably cannot perform any of above actions using these values, since all the important parameters such as \(R_S = Q_S \oplus PWD_U, X_S = W_S \oplus PWD_U, V_U = r_U \oplus h(\sigma_U)\) are safeguarded with \(h(.)\), where \(PID_U = h(ID_U || r_0)\) and \(PWD_U = h(PW_U || r_0)\). \(\mathcal{AE}\) can neither obtain the credentials nor build an authentication request \(< M_1 >\) using the stolen SC due to the unavailability of \(ID_U, PW_U\) and \(BIO_U\). At the same time guessing the \(ID_U, PW_U\) and forging \(BIO_U\) are impractical. Therefore, the proposed protocol can withstand smartcard stolen attacks.

Proposition 4. The proposed protocol is secure against user impersonation attacks.

Proof a. Assume a situation where \(\mathcal{AE}\) possesses a valid SC and wants to gain network access by perpetrating user impersonation attack. If \(\mathcal{AE}\) wants to impersonate a legitimate \(U\), he/she requires to build a login request message \(< M_1 >\). On the other hand, \(\mathcal{AE}\) should undergo login phase before making authentication request. During login phase, \(SC\) computes \(\sigma'_U = \text{Rep}(BIO'_U, \theta_0), r_U = V_U \oplus h(\sigma_U'), PID_U = h(ID_U || r_0), PWD_U = h(PW_U || r_0), W_S = X_S \oplus PWD_U\) and then verifies whether the condition \(C_U = h(ID_U || W_S)\) holds. Unless the \(\mathcal{AE}\) passes the correct credentials, he/she cannot enter into the further phases. Therefore, \(\mathcal{AE}\) certainly requires legitimate credentials for any likewise computations. However, the probability of yielding correct \(ID_U\) and \(PW_U\) is negligible. Though the \(\mathcal{AE}\) performs guessing attacks for \(ID_U\) and \(PW_U\), he/she definitively cannot forge or copy valid \(U\)'s \(BIO_U\).

Proof b. The proposed protocol does not share much personal identifiable information of \(U\) to any \(AS\). During login and mutual authentication phase, \(AS\) can obtain only \(PID_U\) of legitimate \(U\) via \(PID_U = B_{US} \oplus h(SID_1 || N_1 || h(ASK))\). For instance, if any \(AS\) turns as \(\mathcal{AE}\) and wants to impersonate a valid \(U\), he/she still requires \(K_S\) of targeted \(AS\) to construct \(Q_S = h(PID_U || K_S)\). In the proposed protocol, \(Q_S\) value is unique for each \(AS\), where \(K_S = h(SID_1 || ASK)\). Aforementioned constraints prove that our scheme is secure from user impersonation attacks.

Proposition 5. The proposed protocol is secure against application server impersonation attacks.
Proof. Consider a scenario where a registered AS turned as $\mathcal{A}$ captures $< M_1 >$ and tries to impersonate valid AS by responding with computed message $< M_2^e >$. From the captured $M_1$, $\mathcal{A}$ can barely obtain $PID_U = B_{05} \oplus h(SID_A || N_1 || h(ASK))$ and $N_1$ when it is assumed of having $SID_A$. In order to compute the response $M_2^e$, $Q_S$ value of the targeted AS and $U_i$ is a prerequisite since each AS of the proposed protocol holds unique long-term key $K_5$ which is computed based on $SID_A$ as $K_5 = h(SID_A || ASK)$. Thus, $\mathcal{A}$ cannot compute $SK = h(Q_S || N_1 || N_2 || PID_U)$, $ESU = h(SK || SID_A || N_1 || N_2)$ and reply $U_i$. Let’s take another case where $\mathcal{A}$ computes $SK^e = h(Q_S^e || N_1 || N_2 || PID_U)$, $ESU^e = h(SK^e || SID_A || N_1 || N_2)$, and then sends $M_2^e = < ESU^e, N_2 >$ to $SC$. Upon receiving the response, $U_i$ computes $SK = h(Q_S || N_1 || N_2 || PID_U)$, $ESU = h(SK || SID_A || N_1 || N_2)$ and can identify it as a malicious attempt due to the non-equivalence of messages $ESU^e \neq ESU$. It is evident from the above statements that the proposed protocol can withstand application server impersonation attacks.

Proposition 6. The proposed protocol is secure against password guessing attacks.

Proof. $\mathcal{A}$ may try to guess the $PW_U$ using the extracted parameters stored on $SC(X_u, V_u, C_u, T_u, \theta_u, h(\cdot), h(ASK))$ or keep trying to login while guessing the $PW_U$. However, $\mathcal{A}$ cannot validate the guessed $PW_U$ due to non-availability of parameter $r_u$.

On the other hand, $r_f$ value is protected with $U_i$’s $BIO_U$ in the form of $(\sigma_u, \theta_u) = Gen(BIO_U)$, and $V_u = r_f \oplus h(\sigma_U)$ and it is believed to be impractical to forge a valid $U_i$’s $BIO_U$. The $\mathcal{A}$ definitely cannot proceed further without passing correct $BIO_U$ resulting in failure of validating the guessed password using $PID_U = h(ID_U || r_f)$, $PWD_U = h(PW_U || r_f)$, $W_S = X_u \oplus PWD_U$, $C_u = \oplus h(ID_U || W_S)$. In this way, the proposed protocol is secure against password guessing attacks.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>User anonymity and untraceability</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Perfect mutual authentication</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Prevent replay attacks</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Prevent man-in-middle attacks</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Prevent stolen smartcard attacks</td>
<td>❌</td>
<td>❌</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Prevent user impersonation attacks</td>
<td>❌</td>
<td>❌</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Prevent server impersonation attacks</td>
<td>❌</td>
<td>❌</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Prevent insider attacks</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Prevent denial-of-service attacks</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Prevent password guessing attacks</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Prevent clock synchronization problem</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Efficient password changing phase</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Provides user revocation/re-registration phase</td>
<td>❌</td>
<td>❌</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Proposition 7. The proposed protocol is secure against privileged insider attacks.

Proof. During user registration phase of the proposed protocol scenario, $U_i$ does not submit either the plain credentials or the digest of credentials alone to the RS. $U_i$ submits $PID_U = h(ID_U || r_f)$, and $PWD_U = h(PW_U || r_f)$ to $RS$, where $r_f \in Z_{q^*}$ is a random number. Thus, an insider cannot obtain the original credentials of any $U_i$. Additionally, the proposed protocol is designed on the basis of not maintaining password verification tables and the authentication of entities is being done by verifying the accuracy of received messages such as $D_{05} = h(PID_U || Q_S || N_1)$. Therefore, the proposed protocol attains resistance to insider attacks.
Table 3. Comparison of Computation Cost with Other Three-Factor Protocols

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Login &amp;</td>
<td>U_i</td>
<td>8 Th</td>
<td>10 Th</td>
<td>9 Th</td>
<td>7 Th + 3 T_p</td>
<td>8 Th</td>
<td>9 Th</td>
</tr>
<tr>
<td>Authentication</td>
<td>AS</td>
<td>8 Th</td>
<td>7 Th</td>
<td>7 Th</td>
<td>6 Th + 3 T_p</td>
<td>6 Th</td>
<td>6 Th</td>
</tr>
<tr>
<td>Total</td>
<td>RC</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>10 Th + 2 T_p</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>16 Th</td>
<td>17 Th</td>
<td>16 Th</td>
<td>23 Th + 8 T_p</td>
<td>14 Th</td>
<td>15 Th</td>
</tr>
<tr>
<td>Password &amp;</td>
<td>U_i</td>
<td>2 Th</td>
<td>5 Th</td>
<td>5 Th</td>
<td>2 Th</td>
<td>4 Th</td>
<td>6 Th</td>
</tr>
<tr>
<td>Biometrics</td>
<td>RC</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Changing</td>
<td>Total</td>
<td>2 Th</td>
<td>5 Th</td>
<td>5 Th</td>
<td>2 Th</td>
<td>4 Th</td>
<td>6 Th</td>
</tr>
</tbody>
</table>

4. Performance Analysis

This section demonstrates the comparison between the proposed protocol and other related protocols in terms of various aspects such as security, and computational cost. The performance analysis ensures that the proposed protocol is efficient and better in every aspect when compared to other authentication protocols for multi-server architecture.

Functionality comparison: Here, the proposed protocol is compared with the three-factor authentication protocols such as Chuang et al., Mishra et al., Lu et al., He et al., and Wang et al., and is showed in Table 2. It is evident from the Table that except the proposed protocol, all the other three-factor protocols are vulnerable to various security attacks whereas the proposed protocol can prevent user and server impersonation attacks, and also provides perfect user anonymity, user revocation and re-registration phase.

Computational cost comparison: To evaluate the computational cost analysis, we give few notations for the involved actions in all the compared protocols such as $T_h$: Time complexity of a one-way hash function; $T_p$: Time complexity of a point multiplication operation on elliptic curve; $T_f$: Time complexity of encryption or decryption function. To evaluate the computational time analysis, we account $T_h \approx 0.0023ms$, $T_p \approx 2.226ms$, $T_f \approx 0.0046ms$ as reported in [12].

From the Table 3, it is evident that Chuang et al., Mishra et al., Lu et al., He et al., and Wang et al., and the proposed protocol’s login and authentication phase requires the computation complexity of $16 T_h$, $17 T_h$, $16 T_h$, $23 T_h + 8 T_p$, $14 T_h$, and $15 T_h$ respectively. As shown in Table 4, Chuang et al., Mishra et al., Lu et al., He et al., and Wang et al., and the proposed protocol’s login and authentication phase requires the computation time of 0.036ms, 0.039ms, 0.036ms, 17.860ms, 0.032ms, and 0.035ms, respectively. The proposed protocol consumes less computations compared to Chuang et al., Mishra et al., Lu et al., and He et al.’s protocols and more computations compared to Wang et al.’s protocol.

5. Conclusion

This paper has conducted an extensive study of the existing authentication protocols for multi-server environment that are developed based three-factor methodologies. We have reviewed few of the recently proposed protocols and spotted severe security drawbacks due to the flaws in their designs. Thus, we have proposed another protocol on the key point of achieving mutual authentication with key-agreement without sharing decisive personal identifiable information. The proposed protocol is also a three-factor and light-weight protocol. The comparison and analysis sections of this paper proves that the proposed protocol performs better than other related protocol.
Acknowledgment

This work was supported by the BK21 Plus project (SW Human Resource Development Program for Supporting Smart Life) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (21A20131600005).

References


85-95.


Alavalapati Goutham Reddy received his master degree in computer science and engineering from Christ University, India in the year 2013. He is currently a Ph.D. student at School of Computer Science and Engineering at Kyungpook National University, South Korea. His primary research interests revolve around cryptographic authentication protocols and information security. He is a student member of IEEE and ACM.

Eun-Jun Yoon received his Ph.D. degree in computer engineering from Kyungpook National University, South Korea in the year 2006. He is now a professor at Department of Cyber Security, Kyungil University, South Korea. His research interests are cryptography, authentication technologies, smart card security, multimedia security, network security, mobile communications security, and steganography. He has published 75 conference proceedings and 50 journal publications.

Young-Ju Kim received his B.S. degree in computer science and engineering from in the year 2015. He is currently pursuing his M.S. degree at School of Computer Science and Engineering at Kyungpook National University, South Korea. His primary research interests revolve around steganography.

Kee-Young Yoo received his M.S. degree in computer engineering from KAIST, Korea in 1978 and Ph.D. degree in Computer Science from Rensselaer Polytechnic Institute (RPI), U.S.A in 1992. Currently, he is a professor at School of Computer Science and Engineering at Kyungpook National University, South Korea. His area of expertise includes cryptography, steganography, wireless mesh network and RFID security. He is author of more than 200 conference proceedings and 195 journal publications.